Determination of optimal weights of SimBet's routing metrics using entropy weight method

Cite as: AIP Conference Proceedings **2542**, 030007 (2022); https://doi.org/10.1063/5.0103188
Published Online: 10 November 2022

Bambang Soelistijanto

Determination of Optimal Weights of SimBet's Routing Metrics Using Entropy Weight Method

Bambang Soelistijanto[a)]

Department of Informatics, Universitas Sanata Dharma, Yogyakarta, Indonesia

[a)]Corresponding author: b.soelistijanto@usd.ac.id

Abstract. Opportunistic Mobile Networks (OMNs) are an extension of MANETs where the end-to-end transfer delays in these networks are much larger than conventional networks, such as Internet. Consequently, routing algorithms in OMNs only rely on nodes' locally available information when making routing decisions. To improve the delivery performances, most of the OMN routing schemes exploit more than one routing metrics. For instance, SimBet routing considers two social properties of the peer nodes, namely social similarity and betweenness centrality, when selecting better relay nodes. Typically, the algorithm assigns the same weight to both the metrics when making routing decisions. However, our investigation shows that this assignment may lead to the suboptimal performance of the algorithm in real-life OMNs. Therefore, we propose the entropy weight method (EWM) that dynamically assigns the weights of the SimBet's routing metrics for the given node mobility scenario. Finally, simulation results show that the proposed method is able to improve the delivery performances of SimBet routing, in terms of delivery ratio, average delay, and overhead ratio, in several real human movement cases.

INTRODUCTION

Nowadays, opportunistic mobile networks (OMNs) [1] have attracted researchers in mobile communication networks since they offer several advantages over conventional mobile ad-hoc networks (MANETs). For example, an OMN source node can transfer information data to the destination in absence of end-to-end paths between the end nodes. In OMNs, when messages are sent then the routes are dynamically built. Messages are forwarded hop-by-hop from sources to destinations opportunistically when node contacts occur. This message delivery paradigm is known as *store-carry-forward*, and node mobility in OMNs creates chances for message transfers; however, in MANETs node movement is considered as a potential communication disconnection. Moreover, message transfers in OMNs are naturally delay-tolerant, since node contacts occur opportunistically as node mobility is effectively random. The rapid development of mobile devices, such as gadgets, smart phones, and laptops, paves the way for a multitude of opportunities for device encounters. Some possible realizations of OMNs include vehicular networks [2], mobile social networks [3], and animal wildlife monitoring networks [4].

Multi-hop message routing over OMNs possesses substantial challenges: the rapid changes of the networks' topology, the long delay to learn the network state, and the high cost of flooding the global state data, indicate that the conventional MANETs routing strategies requiring global information are costly and suboptimal, since they may depend on obsolete information. As a consequence, most of OMN routing schemes are intuitive heuristics that choose the best message carriers based on nodes' local information. In [5], routing issues in OMNs are formulated as an optimization problem over the time varying graph.

Typically, routing decisions in OMNs are based on node contact statistics. Earlier OMN routing algorithms generally take into account a single routing metric when selecting suitable relay nodes; for example, Label routing [6] uses node community to choose optimal message carriers to the destination. Despite its simplicity, however, considering only one routing metric may fail to accurately estimate the contact probability between a pair of nodes. As a consequence, a number of recent routing algorithms exploit more than one routing metrics to comprehensively describe node relationship; for instance, SimBet routing [7] considers both social similarity and betweenness centrality

The 4th International Conference on Science and Science Education (IConSSE 2021)
AIP Conf. Proc. 2542, 030007-1–030007-6; https://doi.org/10.1063/5.0103188
Published by AIP Publishing. 978-0-7354-4259-7/$30.00

of the peer nodes to select the best relays in the network. Initially, the authors of SimBet assigned the same weight value to both the metrics when calculating node utility. However, our investigation shows that this assignment may lead to the suboptimal performance of SimBet in some realistic mobility scenarios, such as Reality [8], Sassy [9], and Haggle [10]. Based on this issue, we propose the entropy weight method (EWM) [11] to determine optimal weights of the SimBet's routing metrics for a given mobility scenario. Using simulation driven by real node mobility scenarios, we show that the proposed method is able to improve the performance of SimBet, in terms of delivery ratio, average latency and overhead cost.

The rest of the paper is structured as follows. The second section gives an overview of SimBet routing. The third section discusses the entropy weight method used to optimally assign the weights of the SimBet's routing metrics. The performance improvement of SimBet when it applying EWM in real-life OMNs is presented in the fourth section. Finally, the paper is concluded in the last section.

SIMBET ROUTING

As a class of social-aware routing algorithms in OMNs, SimBet routing [7] exploits social properties of nodes when selecting optimal relay nodes in the network. In this algorithm, routing decisions are made based on two social network properties, namely social similarity and betweenness centrality. Theoretically, calculation of both similarity and betweenness centrality needs global knowledge of the social network. However, due to the long transfer delay in OMNs, instantaneous global knowledge of the network is often unavailable to all the network nodes. While the calculation of social similarity can be done based on an ego-centric view, the calculation of betweenness centrality needs global network information. However, the authors of SimBet argued that the computation of ego-centric betweenness is still important to fit a socio-centric betweenness centrality.

Basically, a node forwards messages to the peer having a higher similarity to the destination; however, if both the nodes have no idea about the destination, the messages will be sent to the peer with a higher betweenness centrality. The social similarity contributes more significantly when the messages are already near the destination, while the betweenness centrality gives significant impacts when the sending node is far from the destination. According to SimBet, the social similarity between node *u* and *v* is defined as the number of common encountered nodes between them as follows

$$Sim_u(v) = |N_1(u) \cap N_1(v)| \tag{1}$$

The definition of the socio (global) betweenness centrality of node *u* is the number of geodesic paths between any pair of nodes in the network given as follows

$$C_B(u) = \sum_{\substack{v \neq w \neq u \\ v,w \in nodes}} \frac{g_{v,w}(u)}{g_{v,w}} \tag{2}$$

However, the authors of SimBet defines betweenness centrality in an ego-network view by taking node *v* and *w* only from the neighborhood of node *u* as follows

$$Bet(u) = \sum_{\substack{v \neq w \neq u \\ v,w \in N_1(u)}} \frac{g_{v,w}(u)}{g_{v,w}} \tag{3}$$

Finally, the SimBet utility metric of node *u* for the destination node *v* is given by combining the similarity of *u* and *v* (Eq. 1) and the betweenness of node *u* (Eq. 3) as follows

$$SimBet_u(v) = \alpha * Sim_u(v) + \beta * Bet(u) \tag{4}$$

where $\alpha + \beta = 1$, and α and β are tunable parameters enabling the adjustment of the relative importance of the two metrics. Originally, the authors of SimBet did not discuss thoroughly about the impact of selection of these parameters' values on SimBet's overall delivery performances. Typically, both α and β are assigned to be the same weight value, that is 0.5. However, our experiment shows that this assignment may result in suboptimal performance of SimBet in real-life OMNs, such as Reality [8], Sassy [9], and Haggle [10] contact data traces. Thus, it needs sufficient knowledge in order to set optimal values of α and β for the given mobility scenario. Our approach, however, enables a node to autonomously determine the weight values of SimBet's routing metrics based on the node's locally available information. Furthermore, we apply the entropy weight method (EWM) for determination of weights of the SimBet's routing metrics, and in the following section we discuss the theory of EWM that is applied in SimBet routing.

ENTROPY WEIGHT METHOD

The entropy weight method discussed here is based on the assessment method of Qui *et. al.* [12] with some improvements to suit our case of SimBet. Suppose there are *m* evaluating indicators that evaluate *n* objects. Therefore, an indicator value matrix $X = (x_{ij})_{m\times n}$ is formed as

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \tag{5}$$

In this case, x_{ij} represents the value of routing metric *j*-th of the *i*-th node. Normalization this matrix to get a new matrix $R = (r_{ij})_{m\times n}$, where r_{ij} is the contribution degree of the *j*-th routing metric of the *i*-th node, and is calculated as follows

$$r_{ij} = \frac{x_{ij}}{\sum_{i=1}^{m} x_{ij}} \tag{6}$$

Subsequently, the entropy of *j*-th routing metric is defined as

$$H_j = -k \ \sum_{i=1}^{m} r_{ij} \ln r_{ij} \tag{7}$$

where $k = 1/\ln m$, and $0 \leq H_j \leq 1$. Further, the entropy of the given routing metric j can be used to determine the weight of the metric as follows. We define d_j as the degree of the contribution of each node *i* at the *j*-th routing metric, and it can be shown as $d_j = 1 - H_j$. Eventually, the weight of each routing metric *j* can be calculated as follows

$$w_j = \frac{d_j}{\sum_{j=1}^{n} d_j} \tag{8}$$

Lastly, we now discuss the implementation of the entropy weight method in the SimBet's forwarding strategy. When a node contact occurs between a current node *u* and its peer *v*, for each message *m* with the destination *d* in the node *u*'s buffer, node *u* develops an indicator matrix X containing the similarity and betweenness centrality values of all the previous contacted nodes with respect to the message destination *d*. For example, in (9) we show the matrix X_u of node *u* for the message *m* with the destination *d*, when node *u* has previous contacts with node *a*, *b*, *c*, *e*, *f*, and *g*, and therefore it has the similarity and betweenness centrality values of the contacted nodes with respect to destination *d*. Moreover, if for instance node *f* has not met the destination *d* yet, then $Sim_f(d) = 0$.

$$X_u(d) = \begin{bmatrix} Sim_a(d) & Bet(a) \\ Sim_b(d) & Bet(b) \\ Sim_c(d) & Bet(c) \\ Sim_e(d) & Bet(e) \\ Sim_f(d) & Bet(f) \\ Sim_g(d) & Bet(g) \end{bmatrix} \tag{9}$$

Afterwards, node *u* calculates the values of α and β using equations (6) – (8), and finally determines $SimBet_v(d)$, which is the SimBet's utility value of the peer *v* for the destination *d*. When $SimBet_v(d) > SimBet_u(d)$, then node *u* forwards the message to the peer node *v*. Thus, our method determines the weight values of similarity and betweenness centrality per message in the buffer of the forwarding node *u*. Since in the case of SimBet, matrix X

certainly only has a two-column size, but the row size depending on the number of nodes already encountered, then we can argue that the computation of α and β for each message in the node buffers is still sensible for mobile nodes with limited computing power and resources, e.g., memory and battery.

PERFORMANCE EVALUATION

We now discuss the performance analysis of SimBet with entropy weight method (hereafter, we call it *SimBet-EWM*) compared with the conventional SimBet (we just call it *SimBet*, hereafter). In SimBet, the values of α and β are initially defined before the simulation runs; whereas, in SimBet-EWM the values of both parameters are calculated per message using the entropy weight method when node contacts occur. In this investigation, we use the ONE simulator [13] driven by realistic mobility datasets, namely Haggle [10], Sassy [9], and Reality [8] contact traces. Haggle trace captured the mobility of 41 participants during IEEE Infocomm 2005 at Grand Hyatt Hotel in Miami, USA and lasted for 3 days. Sassy trace, on the other hand, was taken using a mobile sensor network with 27 T-Mote devices carried by people from the University of St. Andrews. The experiment was conducted for 74 days. Finally, Reality trace recorded the activities of 97 students and staffs during 10 months in the MIT campus. For metric evaluations, we consider delivery ratio, delivery latency, and overhead ratio.

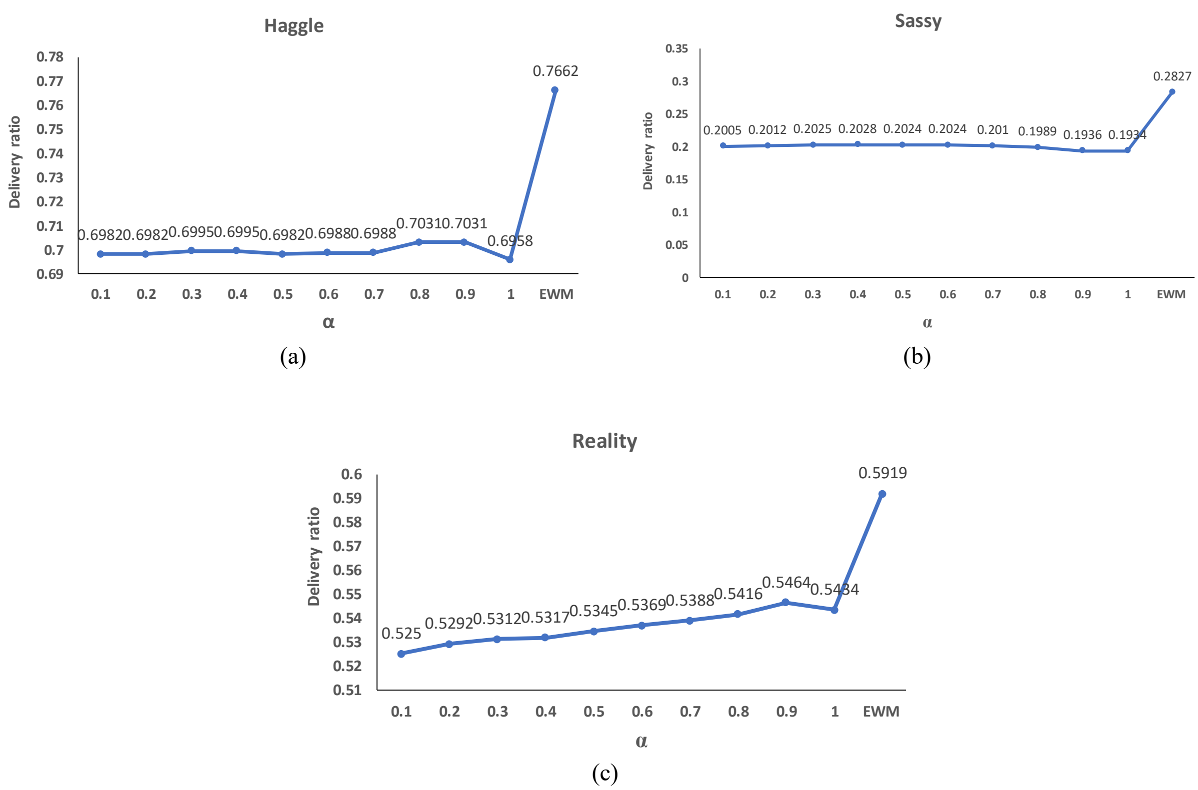
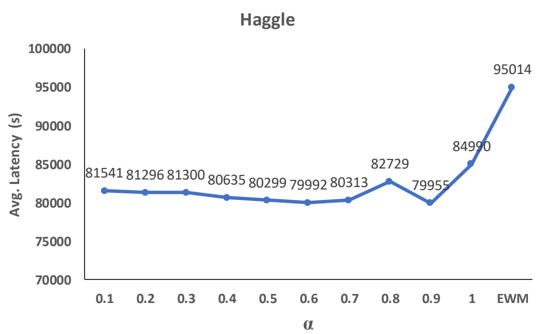


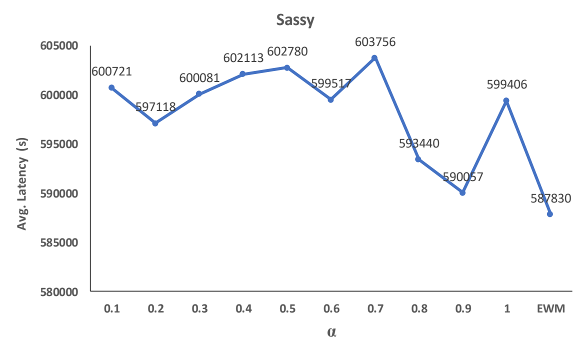
FIGURE 1. Delivery ratio of SimBet vs. SimBet-EWM in three different human mobility scenarios.

In Fig. 1, we depict the delivery success ratio of SimBet-EWM compared with SimBet in three mobility scenarios. For SimBet, we consider various α values in the range of $[0.1,1]$, and $\beta = 1 - \alpha$. From the figure, we can easily see that SimBet-EWM can outperform SimBet in term of delivery success rate in all the given mobility scenarios. In addition, the increase of α (i.e., the forwarding decisions consider more on the social similarity metric) in SimBet gives various impacts in the different mobility scenarios. For example, the increase of α results in an insignificant effect on the message delivery probability in both Haggle and Sassy. However, the increase of α indeed gives a more impact on the delivery ratio in Reality. This is because the social graph of Reality mobility is more clustered than those of Haggle and Sassy, as the students and staffs of MIT typically gathered in certain locations, e.g., laboratories, classroom, and etc., in a relatively long time.

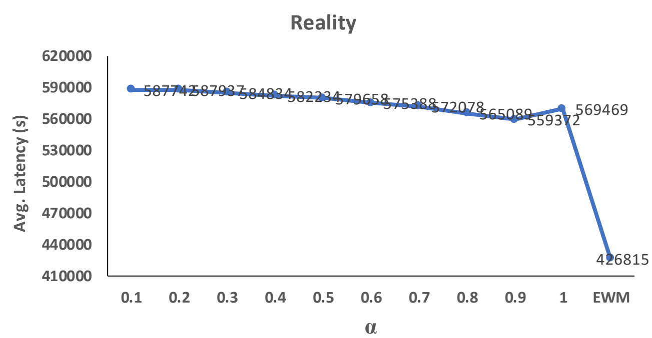
Next, we discuss the performance comparison of SimBet-EWM and SimBet in terms of average delivery latency. We show the delivery delay of SimBet-EWM and SimBet in Haggle, Sassy, and Reality, in Fig. 2. We again notice that SimBet-EWM can significantly reduce the delivery latency of SimBet in Sassy and Reality. However, SimBet-EWM slightly increases the average delay beyond that of SimBet for all values of α. Moreover, in the case of Haggle the increase of α gives a slight impact on the delivery latency. This is because Haggle represents the node mobility scenario in a closed, relatively small area, where all the nodes almost have the same neighbors (thus, having a high similarity).



(a)



(b)



(c)

FIGURE 2. Delivery latency of SimBet vs. SimBet-EWM in three different human mobility scenarios.

Lastly, we now consider overhead ratio performances of SimBet-EWM and SimBet in the aforementioned mobility scenarios. The evaluation metric represents the delivery cost of the routing algorithm, defined as the ratio of total message copies over total messages successfully delivered. We illustrate the delivery cost of SimBet-EWM and SimBet in Haggle, Sassy, and Reality in Fig. 3. It is obvious that SimBet-EWM drastically decreases the delivery cost of SimBet in Reality. This means that the entropy weight method in SimBet-EWM effectively reduces the total message copies in the network, while maintaining the delivery success rate as high as SimBet in Reality. However, in the other mobility scenarios, namely Haggle and Sassy, the performance of SimBet-EWM in terms of delivery cost is roughly similar with that of SimBet for all values of α.

CONCLUSION

In this paper, we improved the performance of SimBet routing by applying the entropy weight method (EWM) to determine the optimal weight values of the two SimBet routing metrics, i.e., social similarity and betweenness centrality. Originally, the weights of both the routing metrics were assigned equally when SimBet made routing decisions. However, we showed that this assignment may result in the suboptimal performance of SimBet in real-life OMNs. Consequently, we proposed the EWM to adaptively calculate the weights of the SimBet's routing metrics for different mobility scenarios. Finally, using simulation driven by real human contact traces, we showed that SimBet-EWM can outperform SimBet, in terms of delivery ratio, average delay, and overhead ratio.

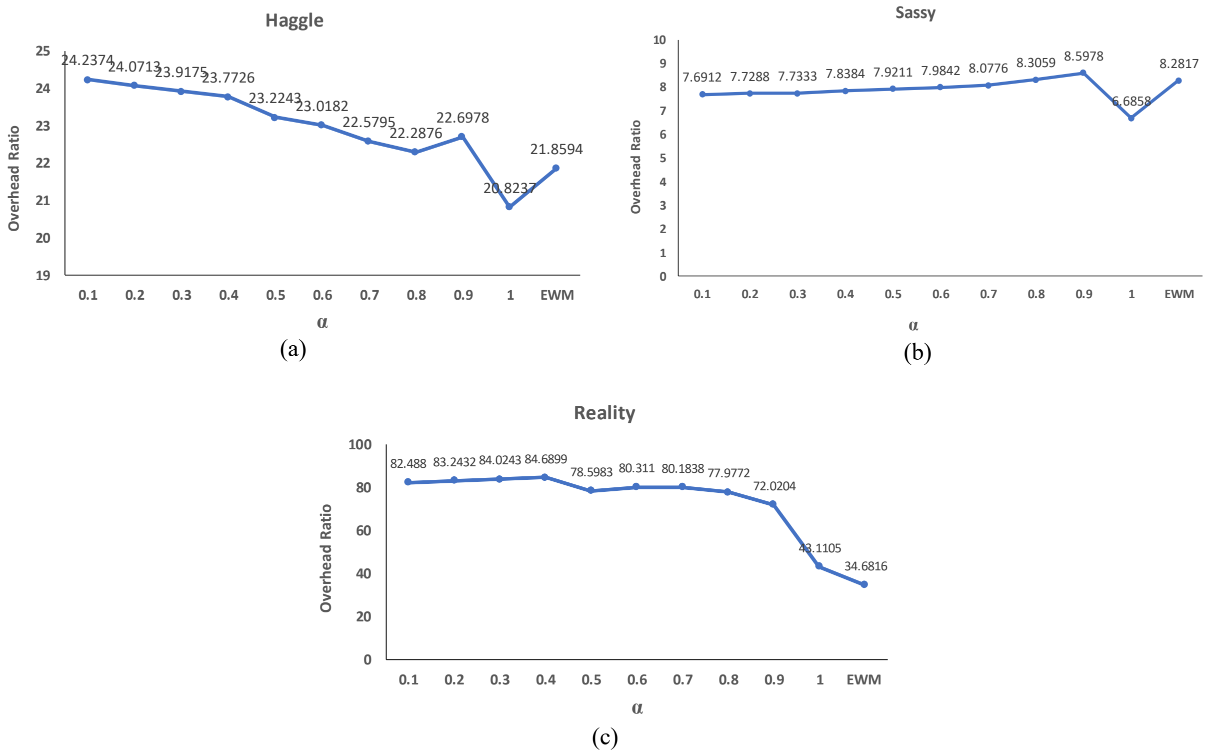


FIGURE 3. Overhead ratio of SimBet vs. SimBet-EWM in three different human mobility scenarios.

ACKNOWLEDGMENTS

This research was partially supported by Sanata Dharma University. We would also like to thank our student, Yustina Lyvita Violita, who provided the Java codes of SimBet-EWM for the ONE simulator.

REFERENCES

1. Z. Liao, L. Liu, and Y. Chen, IEEE Access **8**, 16236–16247 (2020).
2. X. Zhu, Y. Li, D. Jin, and J. Lu, IEEE Trans. Vehicl. Tech. **66**, 7384–7399 (2017).
3. B. Soelistijanto, "Impact of social-aware forwarding on traffic distribution in social opportunistic networks," in Proceedings of the IEEE Region 10 Symposium (IEEE, 2016), pp. 13–18.
4. E. D. Ayele, N. Meratnia, and P.J.M. Havinga, "Towards a new opportunistic IoT network architecture for wildlife monitoring system," in Proceedings of the 9th IFIP International Conference on New Technologies, Mobility, and Security (IEEE, 2018), pp. 1–5.
5. J. Wu, Y. Guo, H. Zhou, L. Shen, and L. Liu, IEEE Access **8**, 18727–18740 (2020).
6. P. Hui, J. Crowcroft, and E. Yoneki, IEEE Trans. Mobile Comp. **10**, 1576–1589 (2011).
7. E. M. Daly and M. Haahr, IEEE Trans. Mobile Comp. **8**, 606–621 (2009).
8. N. Eagle and A. Pentland, Personal and Ubiquitous Comp. **10**, 255–268 (2006).
9. G. Bigwood, D. Rehunathan, M. Bateman, T. Henderson, and S. Bhatti, CRAWDAD dataset st_andrew/sassy (2013), available at http://crawdad.org/st_andrew/sassy.
10. J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "*CRAWDAD dataset Cambridge/Haggle/Infocom (2009),"* available at http://crawdad.cs.dartmouth.edu/cambridge/haggle/ imote/ infocom.
11. L. Liu, J. Zhou, X. An, Y. Zhang, and L. Yang, Expert Syst. App. **37**, 2517–2521 (2010).
12. W. H. Qiu, *Management Decision and Applied Entropy Theory* (China Machine Press, Beijing, 2001).
13. A. Keranen, J. Ott, and T. Karkkainen, "The ONE simulator for DTN protocol evaluation," in Proceedings of the 2nd International Conference on Simulation Tools and Techniques (ICST, 2009), pp. 1–10.